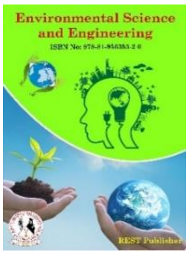
Environmental Science and Engineering
Vol: 2(2), 2023
REST Publisher; ISBN: 978-81-956353-2-0
Website: http://restpublisher.com/book- series/environmental-science-and-engineering/
DOI: https://doi.org/10.46632/ese/2/2/9


Environmental Science and Engineering
ISBN No: 978-81-956353-2-0
REST Publisher

Automatic Extraction of Watershed Model by using Python Programming

*Sunandana Reddy Machireddy
Rajeev Gandhi Memorial College of Engineering & Technology (Autonomous), Nandyal, AP, India
*Corresponding Author Email: machireddynanda@gmail.com

Abstract: *In earlier days, watershed delineation was mainly conducted by the method of hand delineation. Now a days GIS tools are being widely used for delineation of watershed boundary and identifying stream network. The advancements and new techniques in information technologies are making it possible to acquire large-scale spatial data through satellite data, elevation models with help of user-friendly programming. Watershed delineation is a process for creating a boundary that represents the contributing area for a specific control point or water outlet, with the intent of characterization and analysis of portions of a study area. Although many GIS tools and software are available for watershed analysis, there is a need for optimized libraries for client-side and for better illustration of data in interactive environment for hydrological data analysis. In this connection, presently we are developing a model for the automatic delineation of watershed area from Shuttle Radar Topographical Mission (SRTM) digital elevation model by using Python program. The data required by the model are digital elevation and the spatial coordinate of spill way, outlets etc. In this project, we can be developed and demonstrated several watershed delineation techniques on large scale by using Python programming.*

keywords: *Watershed, Hydrological data sets, Elevation Model, Python Programming*

1. INTRODUCTION

Water resource plays a major role in all forms of life. It is also a basic need for any planning and development. Due to cumulative population, hydropower generation, irrigation and industrialization the required amount for water has gone to a large extent. So the delineation of watershed is necessary for hydrologic analysis. Watersheds, also known as basins, which are substantially delineated digitally by GIS on the area of upstream from a stated outlet point. It can be delineated manually on paper as well, but GIS gives faster results. The boundary between two contiguous watersheds is the drainage line and outlet point is the lowest point in elevation at which the water flows out of the area along drainage line. A watershed is a unit of land that distributes water to streams, creeks, or rivers and eventually to lakes and oceans. These land units are also called catchments, drainage basins, and river valleys. Drainage divides or continental divides are ridges that separate watersheds. Watershed delineation is an important concept as watersheds are the unit of study in hydrology. Land cover and land uses within watersheds affect the hydrologic cycle or the flow of water in that area. Defining a watershed boundary is the first step in determining a watershed's characteristics. After a watershed is properly delineated, engineers can calculate the drainage area, slope and land cover. These characteristics along with other factors will affect the time it takes rainwater to travel from the delineation boundary to the downstream outfall. This activity involves delineating a watershed by hand using a topographic map (how engineers used to do it) and then using the USGS Stream Stats site, an online tool engineers use today. Delineation of watershed can also be done by using python programming which gives an accurate results and it can be done for various maps by using few lines of code, which reduces the time for calculation compare to GIS.

Introduction about Hydrology Tools: The ArcGIS Spatial Analyst extension provides a rich set of spatial analysis and modelling tools for both raster (cell-based) and feature (vector) data. The capabilities of Spatial Analyst are broken down into categories or groups of related functionality. Knowing the categories will help you identify which particular tool to use. The table at the end of this section lists all the available toolsets with a description of the capabilities offered by the tools in each. There are several ways to access Spatial Analyst functionality. With geoprocessing, operations in the Spatial Analyst toolbox can be performed through a Tool dialog box, Python (either at an interactive command line interface or with a script), or a Model. Traditional operations and workflows using Map Algebra can also be performed in the Python environment. There is also a Raster Calculator available for entering simple Map Algebra expressions that generate an output raster.

Flow Direction: The output of the Flow Direction tool is an integer raster whose values range from 1 to 255. The values for each direction from the centre. If a cell is lower than its eight neighbours, that cell is given the value of its lowest neighbour, and flow is defined toward this cell. If multiple neighbours have the lowest value, the cell is still given this value, but flow is defined with one of the two methods explained below. This is used to filter out one-cell sinks, which are considered noise. If a cell has the same change in z-value in multiple directions and that cell is part of a sink, the flow direction is referred to as undefined. In such cases, the value for that cell in the output flow direction raster will be the sum of those directions.

Flow Accumulation: The result of Flow Accumulation is a raster of accumulated flow to each cell, as determined by accumulating the weight for all cells that flow into each downslope cell. Cells of undefined flow direction will only receive flow; they will not contribute to any downstream flow. A cell is considered to have an undefined flow direction if its value in the flow direction raster is anything other than 1, 2, 4, 8, 16, 32, 64, or 128. The accumulated flow is based on the number of cells flowing into each cell in the output raster. The current processing cell is not considered in this accumulation. Output cells with a high flow accumulation are areas of concentrated flow and can be used to identify stream channels. Output cells with a flow accumulation of zero are local topographic highs and can be used to identify ridges. If the input flow direction raster is not created with the Flow Direction tool, there is a chance that the defined flow could loop. If the flow direction does loop, Flow Accumulation will go into an infinite loop and never finish. The Flow Accumulation tool does not honour the Compression environment setting. The output raster will always be uncompressed.

Basin: The drainage basins are delineated within the analysis window by identifying ridge lines between basins. The input flow direction raster is analysed to find all sets of connected cells that belong to the same drainage basin. The drainage basins are created by locating the pour points at the edges of the analysis window (where water would pour out of the raster), as well as sinks, then identifying the contributing area above each pour point. This results in a raster of drainage basins. The best results will be obtained if when the input Flow Direction raster was created, the Force all edge cells to flow outward option (FORCE in Python) was enabled. All cells in the raster will belong to a basin, even if that basin is only one cell.

Introduction to Python: Python is a high level, interpreted, general purpose programming language. Its design philosophy emphasizes code readability with use of significant indentation. It is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured, object oriented and functional programming. Guido van Rossum began working on python in the late 1980s as a successor to the ABC programming language and first released it in 1991 as python 2.0 was released in 2000 and introduced new features such as list comprehensions, cycle detecting garbage collection, reference counting, and Unicode support. Python 3.0, released in 2008, was a major revision that is not completely backward-compatible with earlier versions. Python 2 was discontinued with version 2.7.18 in 2020.Now-a-days we are currently using python version 3.10.4. Accurate delineation of a watershed plays an extremely important role in the management of the watershed. The delineated boundaries form the nucleus around which the management efforts such as land use, land change, soil types, geology and river flows are analysed and appropriate conclusions drawn. By using python, it can be done easily with few lines of code in less time.

Applications:

1. To provide necessary inputs for hydrological modelling.
2. Flood prediction modelling and snow melt runoff models.
3. Watershed analysis provides catchment boundaries but also hydrological parameters useful for management programs.

Objectives of Study:

The main objectives of this study are

1. To automate and delineate watershed using python programming.
2. To delineate watershed by using DEM.

Study Area: Nandyal mandal of Kurnool District, Andhra Pradesh, India is selected to demarcate the potential groundwater zones. Geographically Nandyal mandal is located between the latitude from 15° 21 to 15°36 N and the longitude from 78° 20 to 78°34 E. Area of Nandyal mandal is 403.7 sq.km. Elevation of Nandyal mandal is +216.00 m above MSL. Total population of the Nandyal mandal is 2,14,175, spreading across 17 villages. Nandyal is surrounded by hills with the Kundu River on its west, the dense forest of Nallamala Hills to the east and granite mines to the south. The mandal is wealthy in water supply from reservoirs through canals. There is fair climate throughout the year. However, summer can be hot. The average temperature is around 28°C. The Shuttle Radar Topographic Mission data of 1 arc sec spatial resolution data is downloaded from earth explorer of USGS. The same data is used to develop elevation model for the study area. Using this SRTM data, the thematic maps called slope and drainage were extracted with help of geospatial technology.

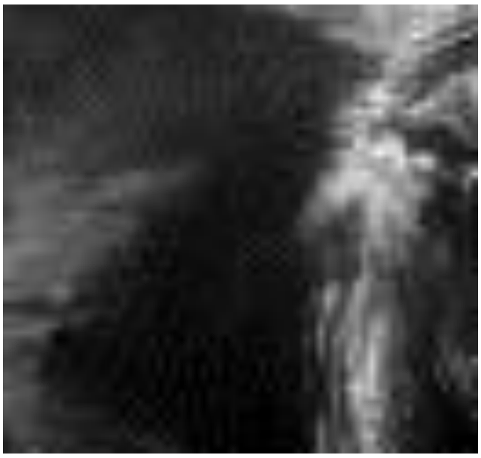


FIGURE 1. DEM of the study area

2. LITERATURE REVIEW

Bal Gopal Guru, Janhabi Meher (2016)

Bal Gopal Guru, Janhabi Meher (2016) Delineation of Mahanadi River Basin by Using GIS and Arc SWAT. Therefore, in the present work Mahanadi river basin lying within Odisha (drainage area approximately 65000 sq. km.) has been delineated in to five subbasins based on the five CWC operated discharge sites in Odisha. In the present work Arc-Swat has been used to delineate the watershed with the help of the (digital elevation model) DEM. It was observed that number of sub-watersheds into which the study area is being depicted relies on number of outlets and density of drainage. The soil data is being found out by the help of the soil map in the grid pattern and was determined by the calculation and viewed in the attribute table after delineation.

Lalitesh Sinha et al (2015)

Lalitesh Sinha et al (2015) did a study on Watershed Delineation of Narmada Basin- Indira Sagar Dam to Maheshwar Dam of Madhya Pradesh. The methodology adopted in this study is watershed delineation using digital elevation model (DEM) and automated watershed tool in

Arc GIS. It is then used to a case study of a Madhya Pradesh where 38 sub basin are determined and used to determine average surface runoff which is the combination of average rainfall and the runoff coefficient. It is concluded that this methodology can be used to generate faster results for estimation of runoff which are important for basin area planning. These results can be effectively used in hydrological modelling, land use planning and watershed studies, reservoir operation and planning.

Parmita bose et al (2009)

Parmita bose et al (2009) have done a study on watershed delineation and flow accumulation calculation using shuttle radar topographic mission (SRTM) data Watershed delineation is very essential for the understanding of geo-environmental condition of an area. In the present study the Shuttle Radar Topographic Mission (SRTM) data has been used for the flow accumulation calculation and watershed basin delineation. The potential drainage were developed based on the 3 arc second Digital Elevation Model (DEM) of SRTM data. Watershed basin has been delineated on the basis of derived stream channels. The results were found very encouraging when it is compared with the drainage basin derived from Survey of India Topographic Sheet.

Sameh et al (2011)

Sameh et al (2011) The present research is focus on evaluating the possibility to delineate catchments from flat and arid areas by means of DTM avoiding hard techniques like river burning or other manual hydrological DTM corrections. Three GIS packages were used (Arc Hydrotools, TNTmips and RiverTools) within two DEM: the 90 m and 30 m SRTM in addition to the ASTER 30 m, the application sample presented by western Iraq desert—Ubaiydh wadis. Where result shows that automated watershed analysis of flat terrains is cannot be done without manual evaluation and correction either by using several seeding points or river burning technique.

3. INTRODUCTION TO PYTHON

Python is a high-level, interpreted, general purpose programming language. Its design philosophy emphasizes code readability with use of significant indentation. It is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured, object oriented and functional programming. Guido van Rossum began working on python in the late 1980s as a successor to the ABC programming language and first released it in 1991 as python 2.0 was released in 2000 and introduced new features such as list comprehensions, cycle detecting garbage collection, reference counting, and Unicode support. Python 3.0, released in 2008, was a major revision that is not completely backward-compatible with earlier versions. Python 2 was discontinued with version 2.7.18 in 2020.Now-a-days we are currently using python version 3.10.4.

Applications of Python

Web Development: We can use Python to develop web applications. It provides libraries to handle internet protocols such as HTML and XML, JSON, Email processing, request, beautiful Soup, Feed parser, etc. One of Python web-framework named Django is used on Instagram.

Scientific and Numeric Application: Python consists of many scientific and mathematical libraries it makes easy to solve complex calculations.

The list of the libraries in the python are:

1. Matplotlib
2. NumPy
3. Pandas
4. SciPy
5. Scikit-learn

3D CAD Application: The CAD (Computer-aided design) is used to design engineering related architecture. It is used to develop the 3D representation of a part of a system. Python can create a 3D CAD application by using the following functionalities.

1. Fandango
2. CAMVOX
3. HeeksCNC
4. AnyCAD
5. RCAM

Image Processing Application: Python contains many libraries that are used to work with the image. The image can be manipulated according to our requirements. Some libraries of image processing are given below.

1. OpenCV
2. Pillow
3. SimpleITK

Business Application: Business Applications differ from standard applications. E-commerce and ERP are an example of a business application. This kind of application requires extensively,
scalability and readability, and Python provides all these features.

Software Development: Python is useful for the software development process. It works as a support language and can be used to build control and management, testing, etc.

Desktop GUI Application: The GUI stands for the Graphical User Interface, which provides a smooth interaction to any application. Python provides a Tk GUI library to develop a user interface. Some popular GUI libraries are given below.

1. Tkinter or Tk
2. wxWidgetM
3. Kivy

Features of Python Programming

Easy-to-learn: Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly. Easy-to-read: Python code is more clearly defined and visible to the eyes. Easy-to-maintain: Python's source code is fairly easy-to-maintaining. A broad standard library: Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh. Interactive Mode: Python has support for an interactive mode which allows interactive testing and debugging of snippets of code. Portable: Python can run on a wide variety of hardware platforms and has the same interface on all platforms. Extendable: You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient. Databases: Python provides interfaces to all major commercial databases. GUI Programming: Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix. Scalable: Python provides a better structure and support for large programs than shell scripting.

Advantages of Python

Easy to Read, Learn and Write: Python is a high-level programming language that has English-like syntax. This makes it easier to read and understand the code. Interpreted Language: Python is an interpreted language which means that Python directly executes the code line by line. In case of any error, it stops further execution and reports back the error which has occurred. Python shows only one error even if the program has multiple errors. This makes debugging easier. Free and Open-Source: Python comes under the OSI approved open-source license. This makes it free to use and distribute. You can download the source code, modify it and even distribute your version of Python. This is useful for organizations that want to modify some specific behavior and use their version for development. Vast Libraries Support: The standard library of Python is huge, you can find almost all the functions needed for your task. So, you don't have to depend on external libraries. But even if you do, a Python package manager (pip) makes things easier to import other great packages from the Python package index (PyPi). Portability: In many languages like C/C++, you need to change your code to run the program on different platforms. That is not the same with Python. You only write once and run it anywhere.

Disadvantages of Python: Slow Speed: We discussed above that Python is an interpreted language and dynamically typed language. The line by line execution of code often leads to slow execution. The dynamic nature of Python is also responsible for the slow speed of Python because it has to do the extra work while executing code. So, Python is not used for purposes where speed is an important aspect of the project. Not Memory Efficient: To provide simplicity to the developer, Python has to do a little trade off. The Python programming language uses a large amount of memory. This can be a disadvantage while building applications when we prefer memory optimization. Weak in Mobile Computing: Python is generally used in server-side programming. We don't get to see Python on the client-side or mobile applications because of the following reasons. Python is not memory efficient and it has slow processing power as compared to other languages. Database Access: Programming in Python is easy and stress-free. But when we are interacting with the database, it lacks behind. Runtime Errors: As we know Python is a dynamically typed language so the data type of a variable can change any time. A variable containing integer number may hold a string in the future, which can lead to Runtime Errors.

Softwares Used:

Software used and the application

1. Anaconda Python 3.8 was utilized for coding.
2. Pysheds library provided the required geo spatial tools.
3. Numerical Python (numpy) was use for array analysis of values.
4. Matplotlib was used for plotting the outcome of the geo spatial analysis.
5. geopandas was used for depicting the geo spatial figures.

Data used:

1. SRTM (Digital Elevation Model)
2. Coordinates of the pour-pointer or outlet of the study area
3. Threshold value to extract the flow accumulation

Modelling:

1. The coding started with the installation of relevant open source softwares like pysheds, geopandas, python etc.
2. The python libraries required for the coding such numerical python, pysheds, geopandas and matplot library were imported in to a Jupyter workspace.
3. The modelling continues with the opening of the raster, detecting and filling or resolving raster depressions, creating the flow direction, delineating catchment and plotting the result after each process and tag was used to indicate comment or explanation.

Watershed delineation program:

A complete set of steps to achieve watershed delineation is

1. Set Inputs
   a. . DEM
   b. . Outlet
   c. . Threshold
2. Flow Direction
3. Flow Accumulation
4. Watershed
5. Catchment
6. Vector Conversion

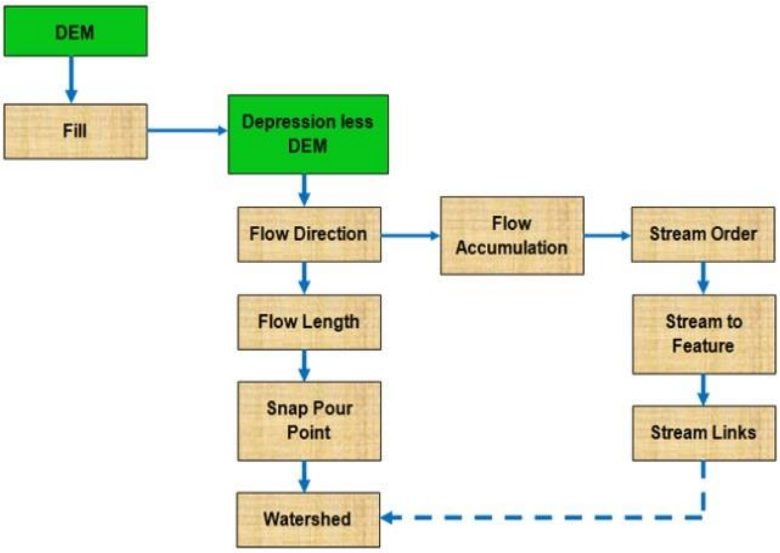
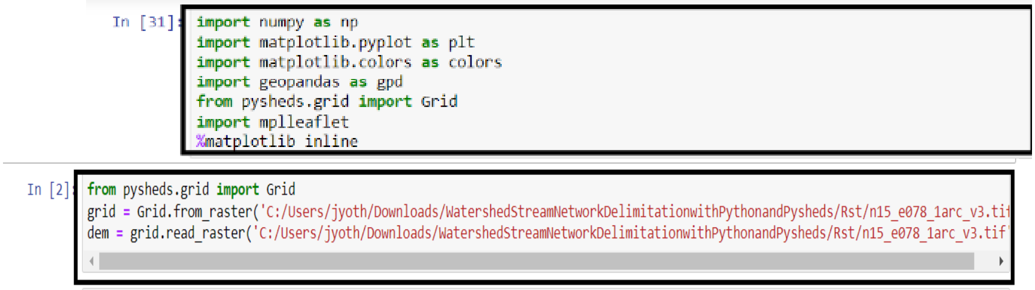


FIGURE 2. Watershed Delineation by Using DEM

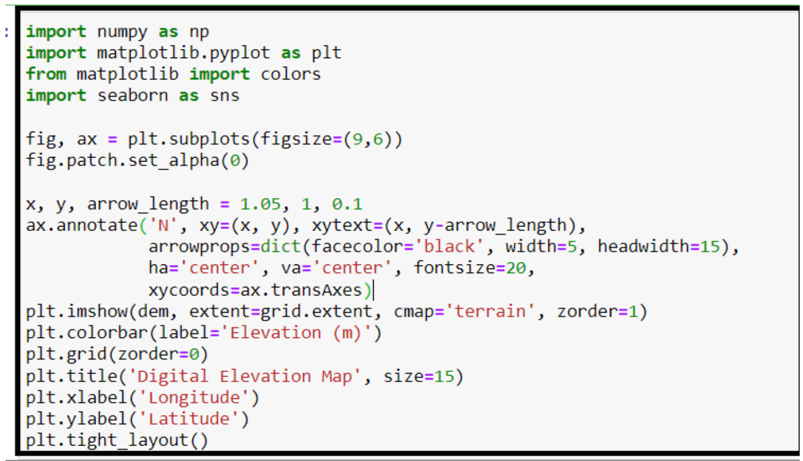
Watershed Delineation by Using DEM: Digital Elevation Models (DEMs) are often used for watershed delineation and data extracted from the delineated watershed relevant required by distributed hydrological models. A watershed is the upslope area that contributes flow generally water to a common outlet as concentrated drainage. It can be part of a larger watershed and can also contain smaller watersheds, called subbasins. The boundaries between watersheds are termed drainage divides. The outlet, or pour point, is the point on the surface at which water flows out of an area. It is the lowest point along the boundary of a watershed. A Watershed can be delineated from a digital elevation model (DEM) by computing the flow direction. This notebook will show you how to delineate Watersheds from a Digital Elevation Model (DEM) based on the python package of pysheds. Pysheds is a Python 3 package designed for watershed delineation and stream network extraction. This library requires a set of advanced data processing and spatial analysis libraries as Numpy, Pandas, Scipy, Scikit-Image, Rasterio and others. DEM is the digital representation of the land surface elevation with respect to fixed point. The elevation data available in analogue form as contour maps which are converted into digital contour files and spatial procedures are applied to interpolate irregular spaced points to regular grid points of elevation values. Afterward a digital representation of surface is developed by which we can find the higher and the lower elevation is called as Digital elevation model.



```
In [31]: import numpy as np
         import matplotlib.pyplot as plt
         import matplotlib.colors as colors
         import geopandas as gpd
         from pysheds.grid import Grid
         import mplleaflet
         %matplotlib inline
```

```
In [2]: from pysheds.grid import Grid
        grid = Grid.from_raster('C:/Users/jyoth/Downloads/WatershedStreamNetworkDelimitationwithPythonandPysheds/Rst/n15_e078_1arc_v3.ti
        dem = grid.read_raster('C:/Users/jyoth/Downloads/WatershedStreamNetworkDelimitationwithPythonandPysheds/Rst/n15_e078_1arc_v3.tif'
```

FIGURE 3. DEM File



```
: import numpy as np
  import matplotlib.pyplot as plt
  from matplotlib import colors
  import seaborn as sns

  fig, ax = plt.subplots(figsize=(9,6))
  fig.patch.set_alpha(0)

  x, y, arrow_length = 1.05, 1, 0.1
  ax.annotate('N', xy=(x, y), xytext=(x, y-arrow_length),
              arrowprops=dict(facecolor='black', width=5, headwidth=15),
              ha='center', va='center', fontsize=20,
              xycoords=ax.transAxes)
  plt.imshow(dem, extent=grid.extent, cmap='terrain', zorder=1)
  plt.colorbar(label='Elevation (m)')
  plt.grid(zorder=0)
  plt.title('Digital Elevation Map', size=15)
  plt.xlabel('Longitude')
  plt.ylabel('Latitude')
  plt.tight_layout()
```

FIGURE 4. Creation of DEM

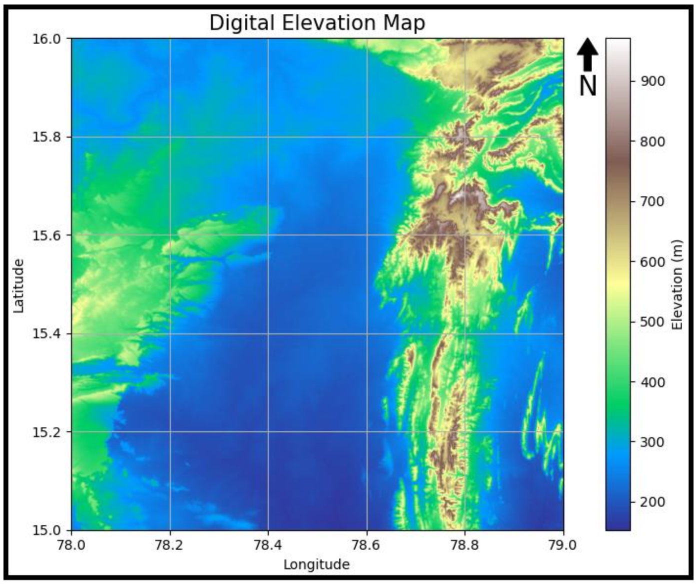


FIGURE 5. Digital Elevation Model

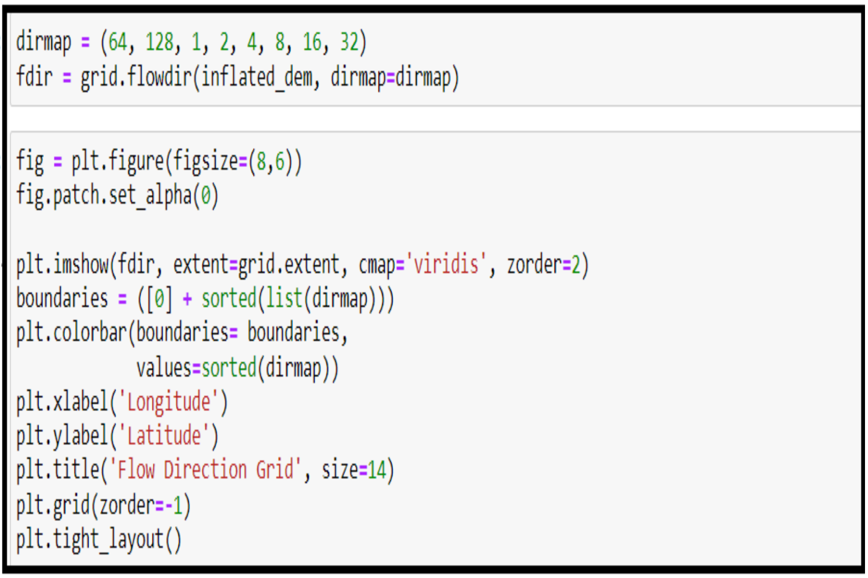
Creation of Flow Direction: It is important to use a DEM with no *depressions* or *sinks*. A sink is a cell that does not have an associated drainage value. Drainage values indicate the direction that water will flow out of the cell, and are assigned during the process of creating a flow direction grid for the landscape. The resulting drainage network depends on finding the 'flow path' of every cell in the grid, so it is important that the fill step is performed prior to creating a flow direction grid.

Compute Flow Direction: A flow direction grid assigns a value to each cell to indicate the direction of flow – that is, the direction that water will flow from that particular cell based on the underlying topography of the landscape.

This is a crucial step in hydrological modelling, as the direction of flow will determine the ultimate destination of the water flowing across the surface of the land. By default, pysheds will compute flow directions using the D8 routing scheme. In this routing mode, each cell is routed to one of eight neighbouring cells based on the direction of steepest descent.

Directional mappings: Cardinal and intercardinal directions are represented by numeric values in the output grid. By default, the ESRI scheme is used:

North: 64
Northeast: 128
East: 1
Southeast: 2
South: 4
Southwest: 8
West: 16
Northwest: 32



```
14]: dirmap = (64, 128, 1, 2, 4, 8, 16, 32)
     fdir = grid.flowdir(inflated_dem, dirmap=dirmap)

30]: fig = plt.figure(figsize=(8,6))
     fig.patch.set_alpha(0)

     plt.imshow(fdir, extent=grid.extent, cmap='viridis', zorder=2)
     boundaries = ([0] + sorted(list(dirmap)))
     plt.colorbar(boundaries= boundaries,
                  values=sorted(dirmap))
     plt.xlabel('Longitude')
     plt.ylabel('Latitude')
     plt.title('Flow Direction Grid', size=14)
     plt.grid(zorder=-1)
     plt.tight_layout()
```

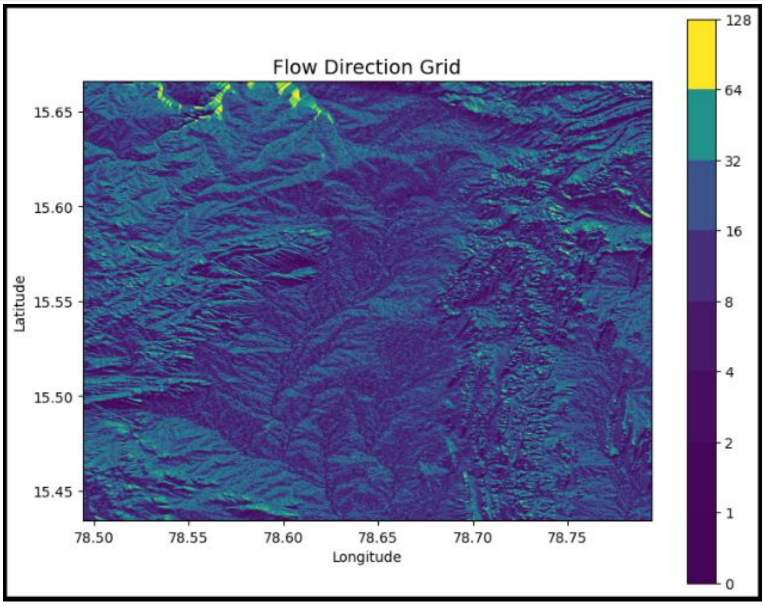
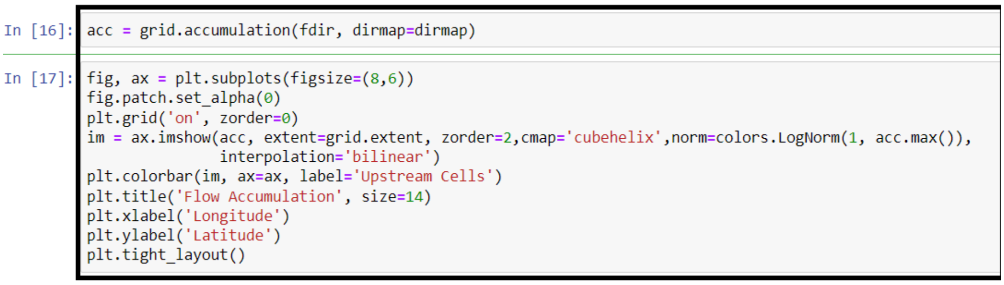


FIGURE 6. Flow Direction

Creation of Flow Accumulation: The Flow Accumulation tool calculates the flow into each cell by identifying the upstream cells that flow into each downslope cell. In other words, each cell's flow accumulation value is determined by the number of upstream cells flowing into it based on landscape topography. The Flow accumulation operation performs a cumulative count of the number of pixels that naturally drain into outlets. The operation can be used to find the drainage pattern of a terrain. As input the operation uses the output map of the Flow direction operation. In the process of simulating runoffs, the flow accumulation is created by calculating the flow direction. To each cell, the flow accumulation is determined by how many cells that flows through that cell; if the flow accumulation value is greater, the area will be easier to form a runoff. The system will calculate the rasters with the weight of 1 by default, if it is not set.



```
In [16]: acc = grid.accumulation(fdir, dirmap=dirmap)

In [17]: fig, ax = plt.subplots(figsize=(8,6))
         fig.patch.set_alpha(0)
         plt.grid('on', zorder=0)
         im = ax.imshow(acc, extent=grid.extent, zorder=2,cmap='cubehelix',norm=colors.LogNorm(1, acc.max()),
                        interpolation='bilinear')
         plt.colorbar(im, ax=ax, label='Upstream Cells')
         plt.title('Flow Accumulation', size=14)
         plt.xlabel('Longitude')
         plt.ylabel('Latitude')
         plt.tight_layout()
```

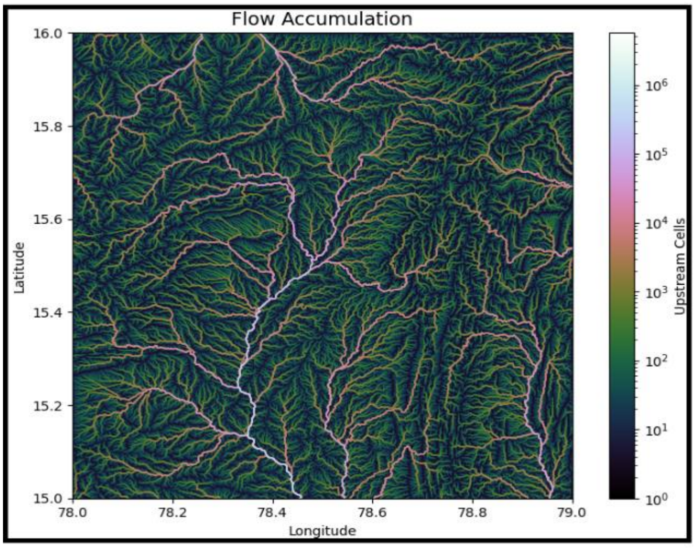
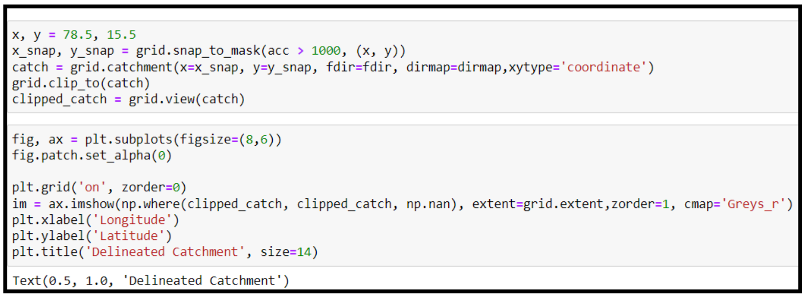


FIGURE 7. Flow Accumulation

Creation of Delineated Catchment: The catchment delineation is used to calculate the direct catchment area or sub-basin from a user selected input location or pour point. The catchment is defined as the land area that contributes runoff to a given stream line. Catchment characteristic derived directly from the geology, soils and land use. Raindrop falling on an edge catchment has a unique path to stream line and thus to being routed down through the stream network. The catchment is representing by polygon feature. To delineate a catchment, first specify a *pour point* (the *outlet* of the catchment). Pour point placement is an important step in the process of watershed delineation. A pour point should exist within an area of high flow accumulation because it is used to calculate the total contributing water flow to that given point. The pour point within the study is selected as x = 78.5 and y = 15.5 to delineate the catchment area at that point. In many cases, you could already know where the locations of your pour points are. If the x and y components of the pour point are spatial coordinates in the grid's spatial reference system, specify xy type='label'.



```
x, y = 78.5, 15.5
x_snap, y_snap = grid.snap_to_mask(acc > 1000, (x, y))
catch = grid.catchment(x=x_snap, y=y_snap, fdir=fdir, dirmap=dirmap,xytype='coordinate')
grid.clip_to(catch)
clipped_catch = grid.view(catch)
```

```
fig, ax = plt.subplots(figsize=(8,6))
fig.patch.set_alpha(0)

plt.grid('on', zorder=0)
im = ax.imshow(np.where(clipped_catch, clipped_catch, np.nan), extent=grid.extent,zorder=1, cmap='Greys_r')
plt.xlabel('Longitude')
plt.ylabel('Latitude')
plt.title('Delineated Catchment', size=14)
```

```
Text(0.5, 1.0, 'Delineated Catchment')
```

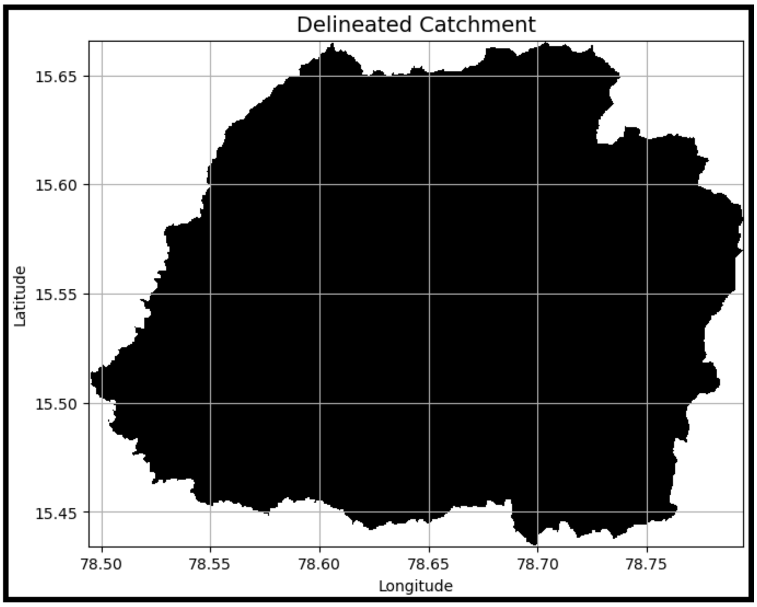
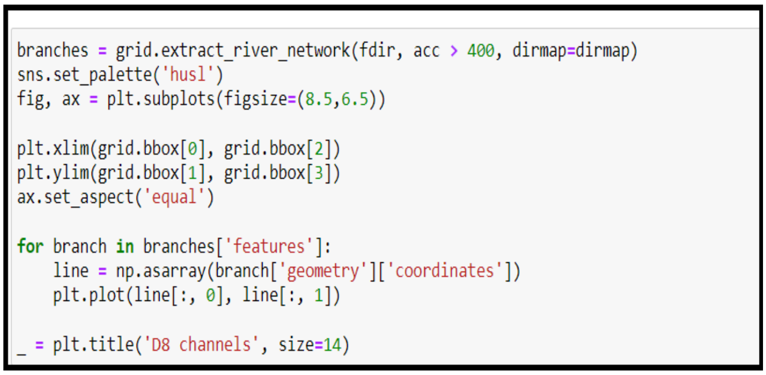


FIGURE 8. Delineated Catchment

Extraction of River Network: The most common method of extracting channel networks from the drainage network is then to specify a critical support area that defines the minimum drainages area required to initiate the channel. Because all cells in a depression less DEM have a flow path to the dataset edge, the pattern formed by highlighting cells with a flow accumulation value greater than some threshold value delineates a fully connected channel network. As the threshold value decreased, more pixels become highlighted and the density of the channel network increases. In practice this threshold value often is selected on the basis of visual similarity between the extracted network and the blue lines depicted on topographic maps. By using threshold number we can extract the drainages. By changing threshold number he network extraction also changes. It is achieved by trail and error method.



```
branches = grid.extract_river_network(fdir, acc > 400, dirmap=dirmap)
sns.set_palette('husl')
fig, ax = plt.subplots(figsize=(8.5,6.5))

plt.xlim(grid.bbox[0], grid.bbox[2])
plt.ylim(grid.bbox[1], grid.bbox[3])
ax.set_aspect('equal')

for branch in branches['features']:
    line = np.asarray(branch['geometry']['coordinates'])
    plt.plot(line[:, 0], line[:, 1])

_ = plt.title('D8 channels', size=14)
```

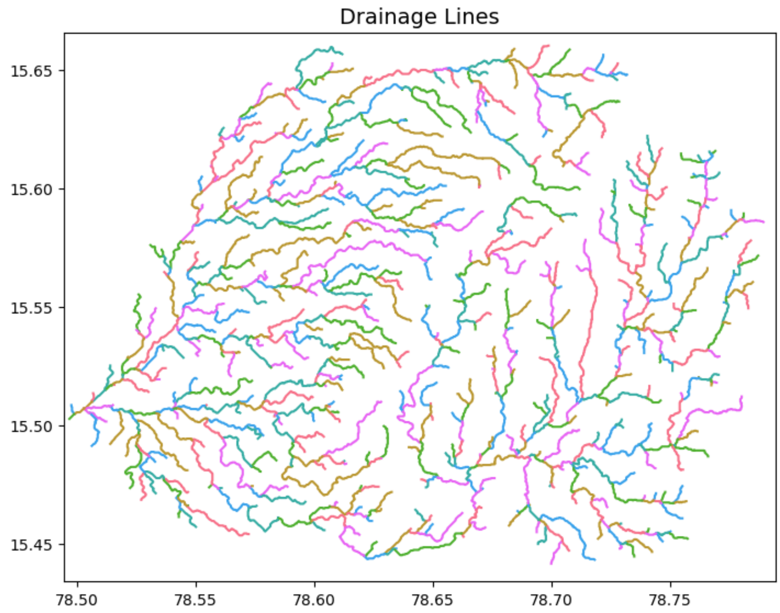
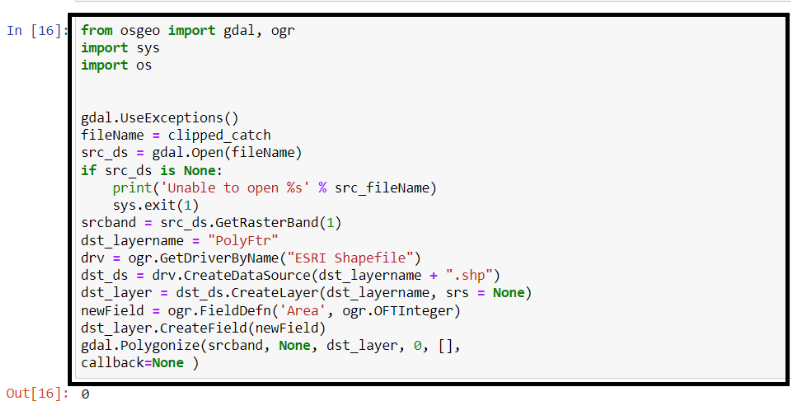


FIGURE 9. D8 Channel



```
In [16]: from osgeo import gdal, ogr
         import sys
         import os


         gdal.UseExceptions()
         fileName = clipped_catch
         src_ds = gdal.Open(fileName)
         if src_ds is None:
             print('Unable to open %s' % src_fileName)
             sys.exit(1)
         srcband = src_ds.GetRasterBand(1)
         dst_layername = "PolyFtr"
         drv = ogr.GetDriverByName("ESRI Shapefile")
         dst_ds = drv.CreateDataSource(dst_layername + ".shp")
         dst_layer = dst_ds.CreateLayer(dst_layername, srs = None)
         newField = ogr.FieldDefn('Area', ogr.OFTInteger)
         dst_layer.CreateField(newField)
         gdal.Polygonize(srcband, None, dst_layer, 0, [],
         callback=None )

Out[16]: 0
```

Conversion of Raster to Vector: By using this code we can convert the raster file to vector layer After converting the raster file to vector file we have to import the shape files of catchment and drainages to calculate the area and length of drainages in QGIS. In QGIS by adding the vector layers of basin and drainages and then by opening the attribute table of selected feature and by using field calculator we can calculate the area and lengths of selected feature.

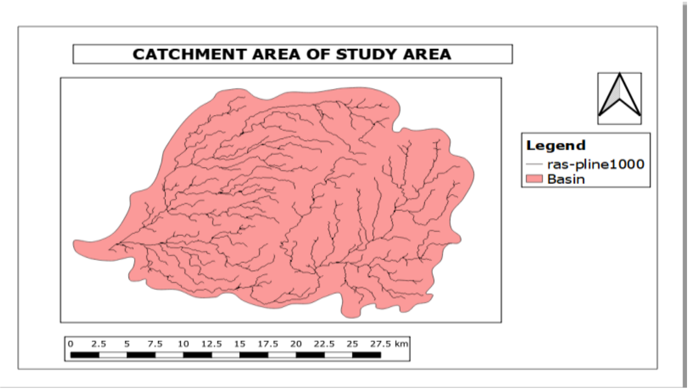


FIGURE 10. Layout of study area

4. RESULT AND DISCUSSIONS

In this project we created watershed delineation by using python programming. To execute the program firstly we have import all the libraries like geopandas, pysheds, matplotlib etc. These are installed by using pip install with library name in environmental variable. After importing all required libraries we have to create a folder consist of our study area which is extracted by USGS. Then we have open jyupter notebook and open the folder and import the study area and assign to one variable. By using the code in previous chapter we can fill the depressions, we can create DEM, Flow Direction, Flow Accumulation, Delineate the study area at one pour point and we can extract the drainage lines by using threshold value. After creating all the files, we have to convert the files to vector and calculate the catchment area and drainage density by using GIS. The catchment area of study area is 628.43 km^2

5. CONCLUSIONS

A common method of locating and delineating the boundaries of watersheds is to use topographic maps following the basic principle that water runs downhill. A topographic map represents the physical features of the land such as hills, valleys, basins, ridges, and channels. The mapping technique used is based on elevation datum (usually mean sea level) and contour intervals commonly. Also by using hydrology tools we can delineate the watershed by using Arc GIS. Compare to other software like QGIS, Arc GIS by using python programming it can be done easily in less time. By using Python, we delineated the study area. From this delineation process we calculated the catchment area and drainage line lengths by using QGIS. And the catchment area as 628438236.83 M^2 and drainage length as 558237.35 M. Finally, from this study we resulted that by using the python we can simply delineated the watershed with few lines of code. The drainage density of study area is Drainage density =Total drain age lengths/Catchment area = 558.23/628.43 = 0.88 According to Smith and Strahler The drainage density 0.88 km/sq.km which is less than 5 as coarse.

REFERENCES

[1]. Al-Khafaji M. S. Al-Sweiti F. H. 2017 Integrated impact of digital elevation model and land cover resolutions on simulated runoff by SWAT model. In: Hydrology And Earth System Sciences Discussions, pp. 1–26. doi:10.5194/hess-2017-653.

[2]. Bal Gopal Guru, Janhabi Meher (2016) Delineation of Mahanadi River Basin by Using GIS and Arc SWAT.

[3]. Cederstrand, J.R., and A. Rea. (1995), Watershed boundaries and digital elevation model of Oklahoma derived from 1:100,000-scale digital topographic maps, U.S. Geological Survey Open-File Report 95-727.

[4]. Chang C. L. 2009 The impact of watershed delineation on hydrology and water quality simulation. Environmental Monitoring and Assessment 148 (1–4), 159–165. doi:10.1007/s10661-007-0147-8.

[5]. Chaplot V. 2005 Impact of DEM mesh size and soil map scale on SWAT runoff, sediment, and NO3–N loads predictions. Journal of Hydrology 312 (1), 207–222. https://doi.org/10.1016/j.jhydrol.20050217.

[6]. Chowdhury M., Hasan M. E. Abdullah-Al-Mamun M. M. 2020 Land use/land cover change assessment of Halda watershed using remote sensing and GIS. The Egyptian Journal of Remote Sensing and Space Science 23 (1), 63–75. https://doi.org/10.1016/j.ejrs.2018.11.003.

[7]. Djokic, D., Z. Ye, and A. Miller. (1997), Efficient Watershed Delineation Using ArcView and Spatial Analyst, Proc. 17th Annual Esri User Conference, San Diego, CA.

[8]. Dr. M. Sunandana Reddy, Mr. J. Sada Sivudu, and Mr. P. Rajesh, Application of Geo-Spatial Technologies in Identification of Groundwater Potential Zones, International Journal for Science and Advance Research In Technology, Vol. 4, Issue.1, 2018.

[9]. Garbrecht, J., and L.W. Martz. (1999), Digital Elevation Model Issues in Water Resources Modelling, Proceedings of the 19th Esri Users Conference, San Diego, CA.

[10]. M.A.E. Bastawesy, R. R. Ali and A. H. Nasr, “The use of remote sensing and GIS for catchments delineation in northwestern coast of Egypt: An assessment of water resource and soil potential”, The Egyptian Journal of Remote Sensing and Space Science, Vol. 11, 2008, pp.3-16.

[11]. Machireddy, S.R., Delineation of groundwater potential zones in South East part of Anantapur District using remote sensing and GIS applications. Sustain. Water Resour. Manag. 5, 1695–1709 (2019). https://doi.org/10.1007/s40899-019-00324-3

[12]. Machireddy, S.R., Assessment and distribution of groundwater quality using Water Quality Index and geospatial technology in Vempalli Mandal of Andhra Pradesh, India. Sustain. Water Resour. Manag. 9, 51 (2023). https://doi.org/10.1007/s40899-023-00828-z

[13]. N. Donia, "Application of Remotely Sensed Imagery to Watershed Analysis; A case Study of Lake Karoun Catchment, Egypt," Thirteen International Water Technology Conference, IWTC, Hurghada, 2009, pp 1035-1049.

[14]. Olivera, F., and D.R. Maidment. (1999), GIS Tools for HMS Modelling Support, Proceedings of the 19th Esri Users Conference, San Diego, CA.

[15]. Perez, A. (1999), Source Water Protection Project: A comparison of Watershed Delineation Methods in ARC/INFO and ArcView, Proceedings of the 19th Esri Users Conference, San Diego, CA (this issue).

[16]. Srivastava, M. (2017) Land Use Classification and Watershed Analysis of Assi River, Varanasi, Uttar Pradesh, India. Journal of Advanced Research in Construction Urban Architecture. 02.32-37. 10.24321/2456.9925.201701.

[17]. Trisakti, B., and Carolita, I. (2010) Comparison Result of DEM Generated from ASTER Stereo Data and SRTM. International Journal of Remote Sensing and Earth Sciences (IJRe- SES) 4(1), 94–101.